



Proyecto Muebleria

Tomas Martinez



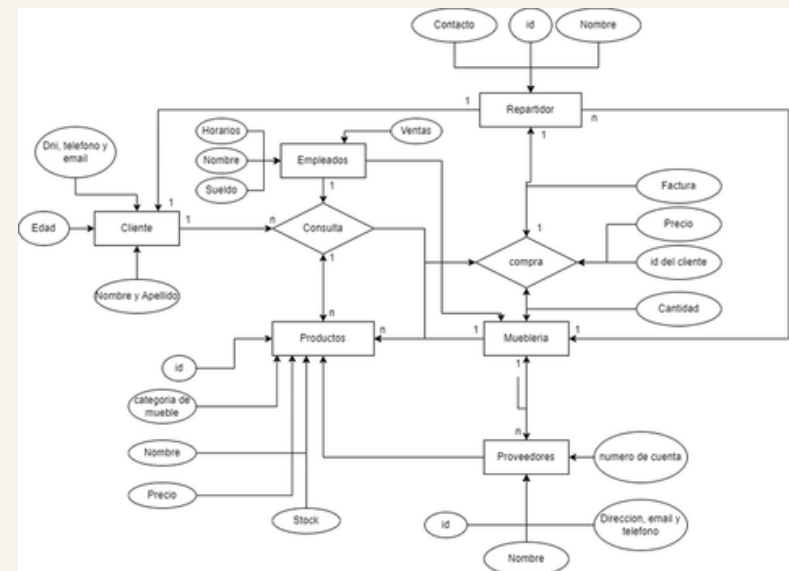
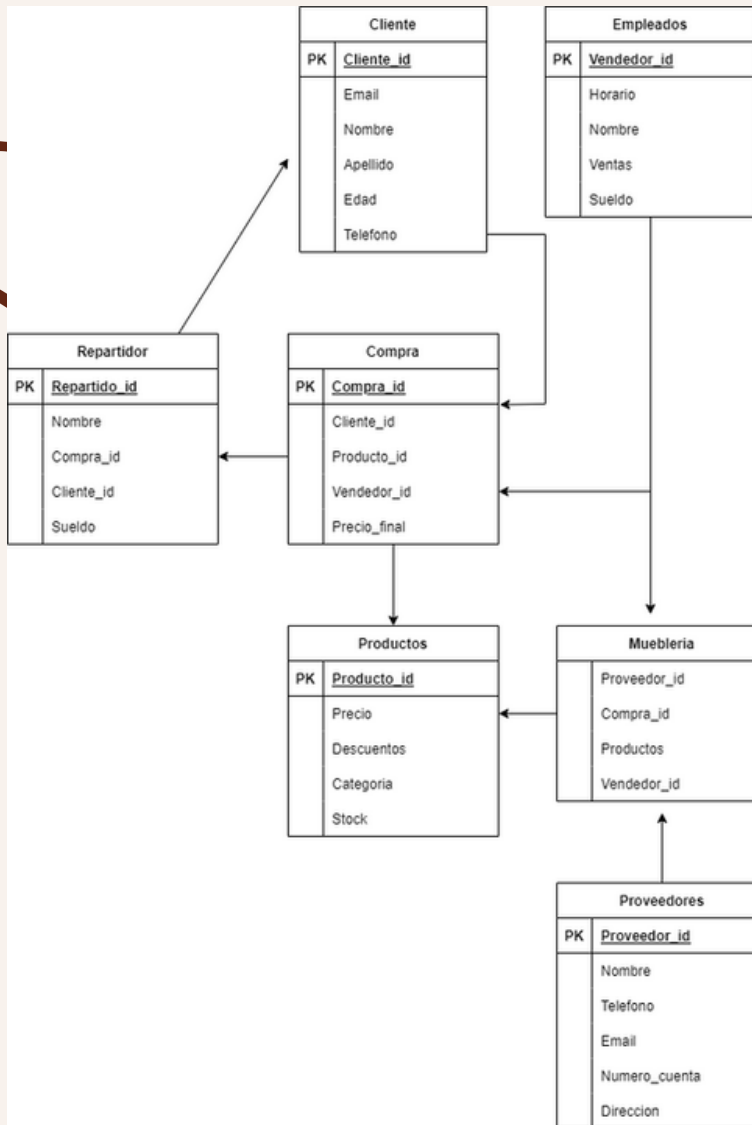
Objetivo

Esta Database tiene como objetivo poder almacenar todos los datos que puede llegar a tener una muebleria, como insertar datos y eliminarlos según sea necesario.



DER

Las siguientes imagenes muestran el diagrama de entidad-relacion de la muebleria, de la cual me baso al momento de crear, actualizar, mejorar o eliminar datos u/o tablas.



Creacion de tablas

Para crear la base de dato y las tablas, hay que copiar y pegar las lineas de codigo de creacion de tablas que se encuentran en la carpeta llamada Data, en el siguiente repositorio.

https://github.com/MarioTomasMP/MySQL_Muebleria/blob/main/Data/Creacion%20de%20tablas.sql

```
1 DROP DATABASE IF EXISTS muebleria_martinez;
2 CREATE DATABASE IF NOT EXISTS muebleria_martinez;
3
4 USE muebleria_martinez;
5
6 DROP TABLE IF EXISTS Cliente;
7 DROP TABLE IF EXISTS Empleado;
8 DROP TABLE IF EXISTS Stock;
9 DROP TABLE IF EXISTS Producto;
10 DROP TABLE IF EXISTS Factura;
11 DROP TABLE IF EXISTS Proveedor;
12 DROP TABLE IF EXISTS Distribuidor;
13 DROP TABLE IF EXISTS Tienda;
14
15 CREATE TABLE IF NOT EXISTS Cliente(
16 id_cliente INT NOT NULL UNIQUE AUTO_INCREMENT PRIMARY KEY,
17 nombre_cliente TEXT NOT NULL,
18 email_cliente VARCHAR(100) NOT NULL,
19 edad_cliente INT NOT NULL,
20 telefono_cliente VARCHAR(30) NOT NULL
21 );
22
23 CREATE TABLE IF NOT EXISTS Cliente(
24 id_cliente INT NOT NULL UNIQUE AUTO_INCREMENT PRIMARY KEY,
25 nombre_cliente VARCHAR(50) NOT NULL,
26 email_cliente VARCHAR(100) NOT NULL,
27 edad_cliente INT NOT NULL,
28 telefono_cliente VARCHAR(30) NOT NULL,
29 direccion_cliente VARCHAR(100) NOT NULL
30 );
31
32 CREATE TABLE IF NOT EXISTS Empleado(
33 id_empleado INT NOT NULL UNIQUE AUTO_INCREMENT PRIMARY KEY,
34 nombre_empleado VARCHAR(50) NOT NULL,
35 salario_empleado FLOAT NOT NULL,
36 telefono_empleado VARCHAR(30) NOT NULL
37 );
38
39 CREATE TABLE IF NOT EXISTS Proveedor(
40 id_proveedor INT NOT NULL UNIQUE AUTO_INCREMENT PRIMARY KEY,
41 nombre_proveedor VARCHAR(50) NOT NULL,
42 telefono_proveedor VARCHAR(30) NOT NULL,
43 direccion_proveedor VARCHAR(100) NOT NULL,
44 email_proveedor VARCHAR(100) NOT NULL,
45 numero_cuenta_proveedor VARCHAR(100)
46 );
47
48 CREATE TABLE IF NOT EXISTS Producto(
49 id_producto INT NOT NULL UNIQUE AUTO_INCREMENT PRIMARY KEY,
50 nombre_producto VARCHAR(50) NOT NULL,
51 precio_producto FLOAT NOT NULL,
52 descuento_producto INT NOT NULL,
53 categoria_producto VARCHAR(100) NOT NULL
54 );
55
56 CREATE TABLE IF NOT EXISTS Factura(
57 id_factura INT NOT NULL UNIQUE AUTO_INCREMENT PRIMARY KEY,
58 preciofinal_factura FLOAT NOT NULL,
59 fecha_factura DATETIME NOT NULL,
60 id_producto INT NOT NULL,
61 id_empleado INT NOT NULL,
62 id_cliente INT NOT NULL,
63 CONSTRAINT FK_Factura_Producto FOREIGN KEY (id_producto) REFERENCES Producto(id_producto),
64 CONSTRAINT FK_Factura_Empleado FOREIGN KEY (id_empleado) REFERENCES Empleado(id_empleado),
65 CONSTRAINT FK_Factura_Cliente FOREIGN KEY (id_cliente) REFERENCES Cliente(id_cliente);
```

Insercion de datos

Truncamiento

Antes de insertar cualquier dato, insertar la siguiente linea de codigo en una hoja de sql y ejecutala.

```
USE muebleria_martinez;  
SET FOREIGN_KEY_CHECKS=0;  
TRUNCATE 'nombre de la tabla a la  
que desea cargar sus datos';  
SET FOREIGN_KEY_CHECKS=1;
```

Copiar datos

Una vez truncado, en otra hoja sql pegar el listado de cosigos de la carpeta Data llamada Insertar datos, que se encuentra en el repositorio antes compratido.

Cargar datos

Una vez realizado los dos anteriores pasos, ejecutar y llamar con el select * from cualquier tabla en la que se hayan cargado sus datos, y deberias poder ver todo cargado correctamente.

```

USE muebleria_martinez;

USE 'muebleria_martinez';
DROP function IF EXISTS 'sumar_productos_x_categoria';

USE 'muebleria_martinez';
DROP function IF EXISTS 'muebleria_martinez`.`sumar_productos_x_categoria`;
;

DELIMITER $$
USE 'muebleria_martinez'$$
CREATE DEFINER='root'@'localhost' FUNCTION `sumar_productos_x_categoria`(categoria VARCHAR(100)) RETURNS int
    READS SQL DATA
BEGIN
    DECLARE productos_totales INT;
    SELECT count(*)
        INTO productos_totales
    FROM producto
    WHERE categoria_producto = categoria;
    RETURN productos_totales;
END$$

DELIMITER ;
;

DROP function IF EXISTS 'muebleria_martinez`.`estado_stock`;
;

DELIMITER $$
USE 'muebleria_martinez'$$
CREATE DEFINER='root'@'localhost' FUNCTION `estado_stock`(stock_total INT) RETURNS varchar(30) CHARSET utf8mb4
    NO SQL
BEGIN
    DECLARE estado_stock VARCHAR(30);
    SET estado_stock = 'UNDEFINED';
    IF stock_total > 20 THEN
        SET estado_stock = 'SUFICIENTE';
    ELSEIF stock_total >= 15 THEN
        SET estado_stock = 'BUENO';
    ELSEIF stock_total >= 10 THEN
        SET estado_stock = 'REGULAR';
    ELSEIF stock_total >= 5 THEN
        SET estado_stock = 'BAJO';
    ELSEIF stock_total >= 1 THEN
        SET estado_stock = 'INSUFICIENTE';
    END IF;

```

Funciones

Para utilizar las funciones, entrar en la carpeta llamada funciones, copia y pega todos los datos y luego ejecuta.

FUNCIONES Y STORED PROCEDURES

para usar las funciones, ir al repositorio donde dice funciones y stored procedures dentro de la carpeta DATA.

Uno de los stored procedures es para hacer un pedido, el otro sirve para poder ver todos los productos, de manera organizada según el parámetro que se pida.

Las funciones que cree sirven para llamar un cliente al azar, un empleado, uno o varios productos, para sacar el total de stock, para ver cuantos productos hay por categoría y saber el estado del stock.



```
USE muebleria_martinez;

SET @@autocommit = 0;

START TRANSACTION;
DELETE FROM empleado WHERE id_empleado = 2 AND nombre_empleado = 'Maria Dominguez';

ROLLBACK;

COMMIT;
SET @@autocommit = 1;

START TRANSACTION;
SELECT * FROM factura;
SELECT * FROM stock JOIN producto WHERE id_stock = id_producto;
INSERT INTO factura VALUES
(1, 87501, '2022-08-01 10:28:50', 1, 3, 4);

savepoint primera_factura;
```

TCL

Para el proyecto cree 2 tcl, uno en el cual se puede eliminar de la lista de empleados, por si se despidiera alguno y el otro con el cual se crearia una factura.

El codigo se encuentra en el repositorio, en la carpeta llamada tcl.

Backups

Dentro de la carpeta Data se encuentra la carpeta backups en la cual se puede encontrar todos los backups de la muebleria, ya sea el completo, como las tablas de manera individual.

Para usar los archivos tenes que entrar en el siguiente link al repositorio
https://github.com/MarioTomasMP/MySQL_Muebleria/tree/main/Data

